

**A LARGE SCALE SOFTWARE SYSTEM FOR
SIMULATION AND DESIGN OPTIMIZATION OF MECHANICAL SYSTEMS**

**Bernhard Dopker
and
Edward J. Haug
Center for Simulation and Design Optimization
and
Department of Mechanical Engineering
College of Engineering
The University of Iowa**

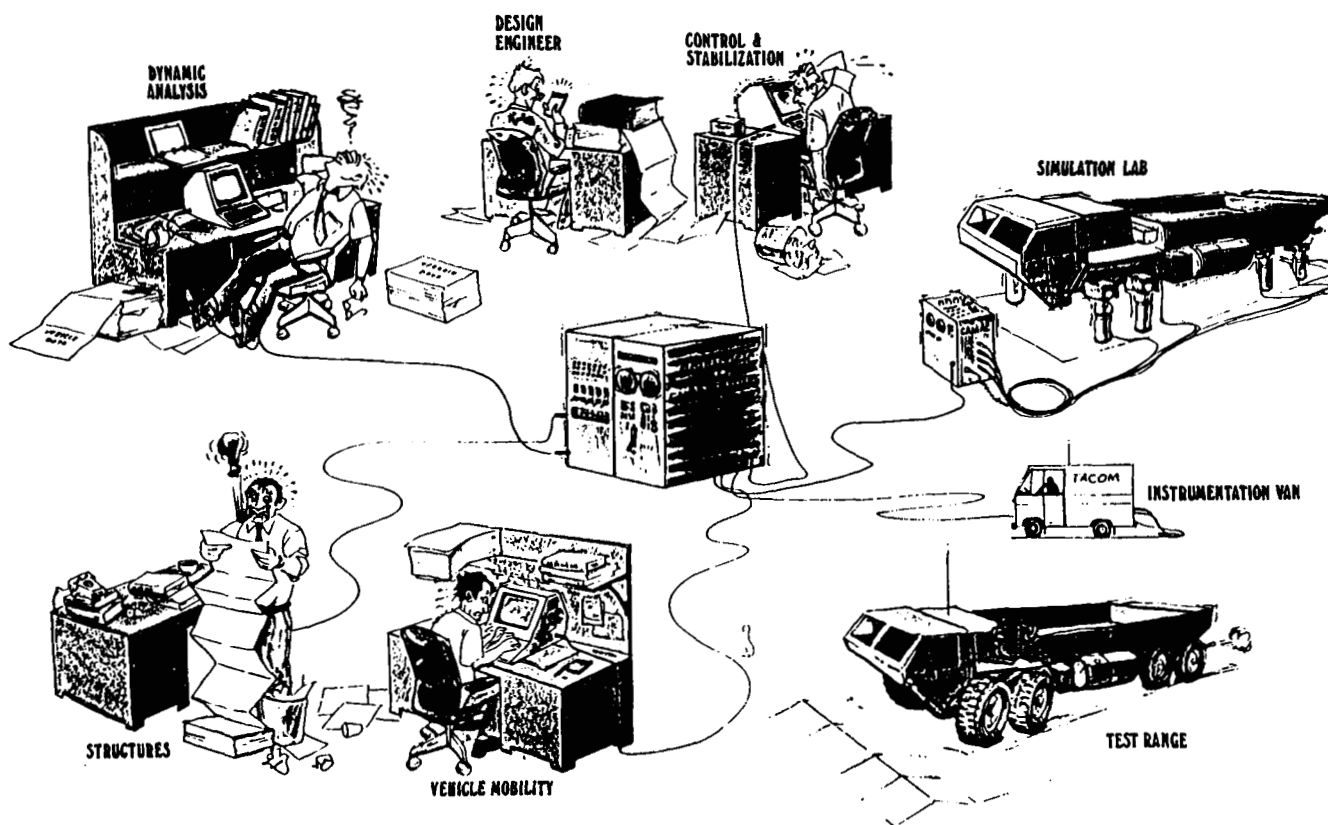
PRECEDING PAGE BLANK NOT FILMED

INTRODUCTION

While the individual technologies of computers, computer graphics, and computational mechanics are well developed, these technologies have not yet been adequately exploited in support of mechanical system simulation and design optimization. Utilization of advanced computational tools by sophisticated development teams in the aerospace industry has demonstrated their potential, but they have not yet been brought to bear in mechanical system design environments, such as military vehicles. Mechanical design teams that currently fabricate and test prototypes should be able to carry out high resolution computer simulations long before committing to hardware. To meet this need and realize enormous cost and time savings, advanced methods and software must be developed and integrated to fully utilize emerging supercomputer and parallel processor architecture, computer graphics for communication, and data transfer between advanced large scale analysis programs.

Major developments have occurred in specialized analysis software, such as finite element structural analysis codes, kinematic and dynamic analysis codes, and control simulation codes. While vendors of CAD and CAE systems are beginning to include support for some specialized analysis software, the potential that exists for large scale interdisciplinary simulation and design optimization support to mechanical system design is virtually untapped. Significant research is contributing to disciplines that are required to meet the needs of mechanical system simulation and design optimization. Unfortunately, most of the research is being carried out by specialists with little or no communication among related disciplines. A simulation and design optimization software system is needed to accelerate interdisciplinary research and development (ref. 1).

CURRENT SIMULATION AIDS TO DESIGN



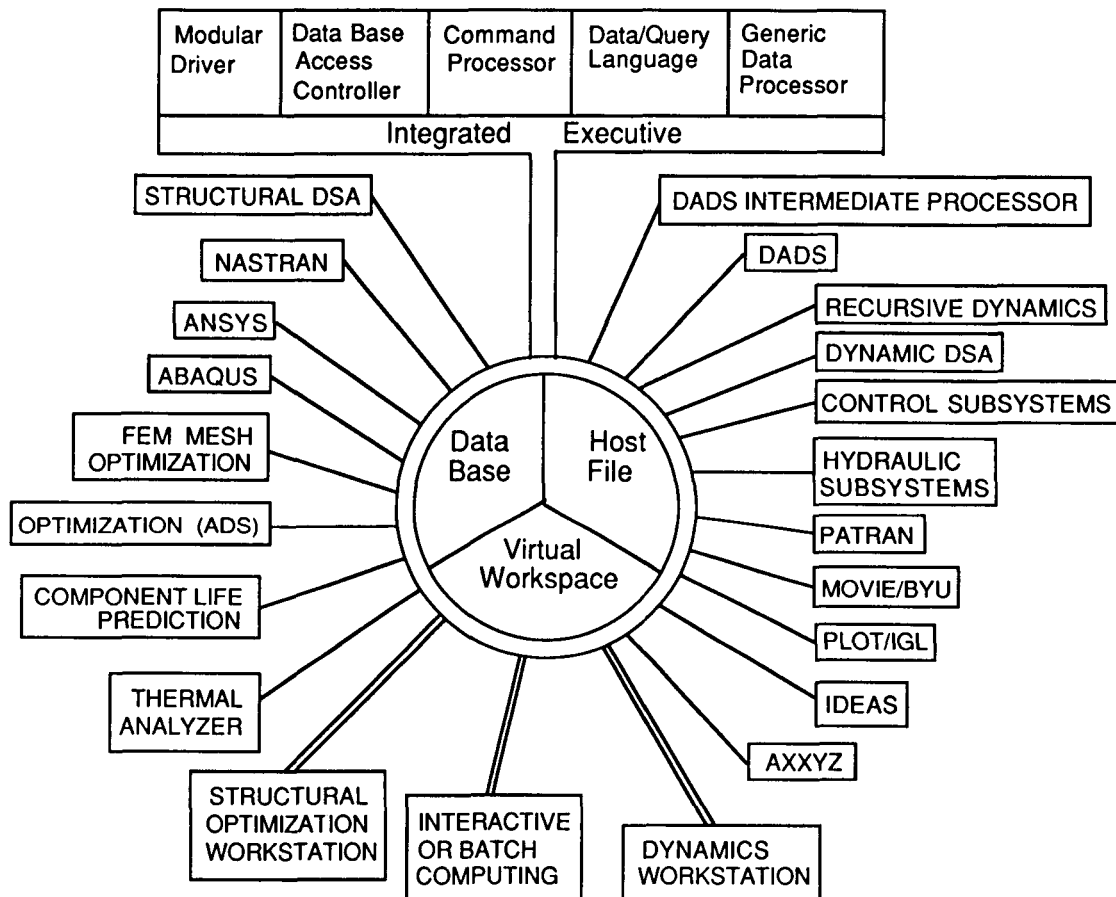
ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

A SOFTWARE TEST BED

An NSF* Industry/University Cooperative Research Center was formed in 1987 to develop advanced mechanical system simulation methods and to implement them into a national research software system. An integrated software system for simulation and design optimization, including software such as finite element codes, kinematics and dynamics codes, control simulation codes, graphics based CAD/CAE codes, and design sensitivity analysis and optimization codes is being implemented. Emerging computer architectures, simulation methods, and data base management systems are being utilized. The goal is integration of discipline oriented software into a data base and command language system that permits research and development teams to carry out their work in an interdisciplinary environment.

This integrated software environment will, by necessity, be based on an underlying computational environment that consists of a network of heterogeneous computing elements. These may include personal computers, workstations, mainframe systems, parallel processors, supercomputers, and a variety of specialized servers for support of data base, graphics, and artificial intelligence activities. Thus, the software support system must be capable of supporting the integration of applications that consist of pieces running on different hardware platforms and under different operating system interfaces. This integration should be compatible with emerging standards for network computing (refs. 2, 3, and 4).

A National Software System For Mechanical System Simulation And Design Optimization



*National Science Foundation

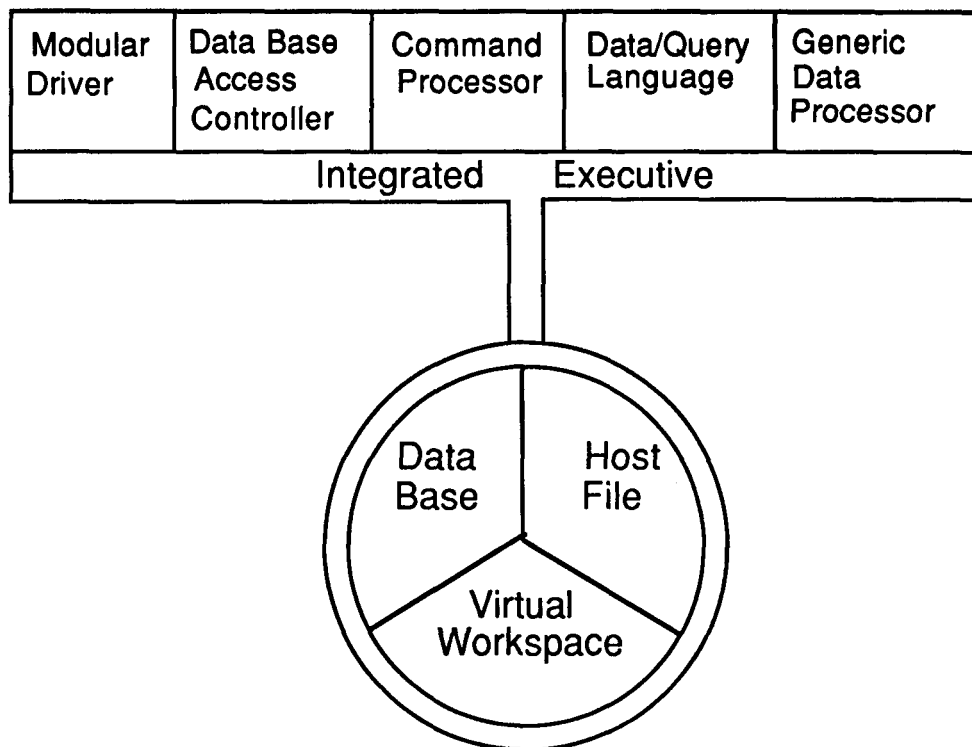
THE INTEGRATED ANALYSIS CAPABILITY

As the basis for integrating the application codes, the Integrated Analysis Capability (IAC) developed by Boeing for NASA/Goddard, the SDI Office, and the Air Force is utilized (refs. 5 and 6). The objective of the IAC has been to provide a highly effective, interactive analysis tool for integrated, interdisciplinary analysis and design of large structures. Having the engineer-in-the-loop was an important design consideration. Emphasis has been placed on the capability to integrate new application programs into a uniform framework. The IAC has been focused on the technical disciplines of structures (both statics and dynamics), controls and thermodynamics. The Analysis Capability Executive (ACE) module, shown at the top of the figure, relies on the same set of standard IAC utilities that are documented and available for use by all modules in the system.

ACE contains a general driver code to execute other application modules; an input command processor that handles language syntax, prompting, help information, etc.; an extensive data/query management capability that allows the user to create, manipulate, and evaluate various types of data; and a generic data processor that allows the interface between the IAC and "foreign" module character formatted data. ACE is independent from any requirements of a particular application module. Multiple concurrent users may interface with IAC to perform both interactive and batch tasks.

The data base utilized by the IAC consists of three parts: (1) the IAC structured data base that allows for storage of data files in a structured form and provides some mathematical operations such as matrix additions and spreadsheet computations; (2) the IAC virtual workspace that allows storage of structured data temporarily and allows the same mathematical operations as on the IAC structured data base; and (3) the host directory of the computer operating system.

The IAC Database and Command Processor

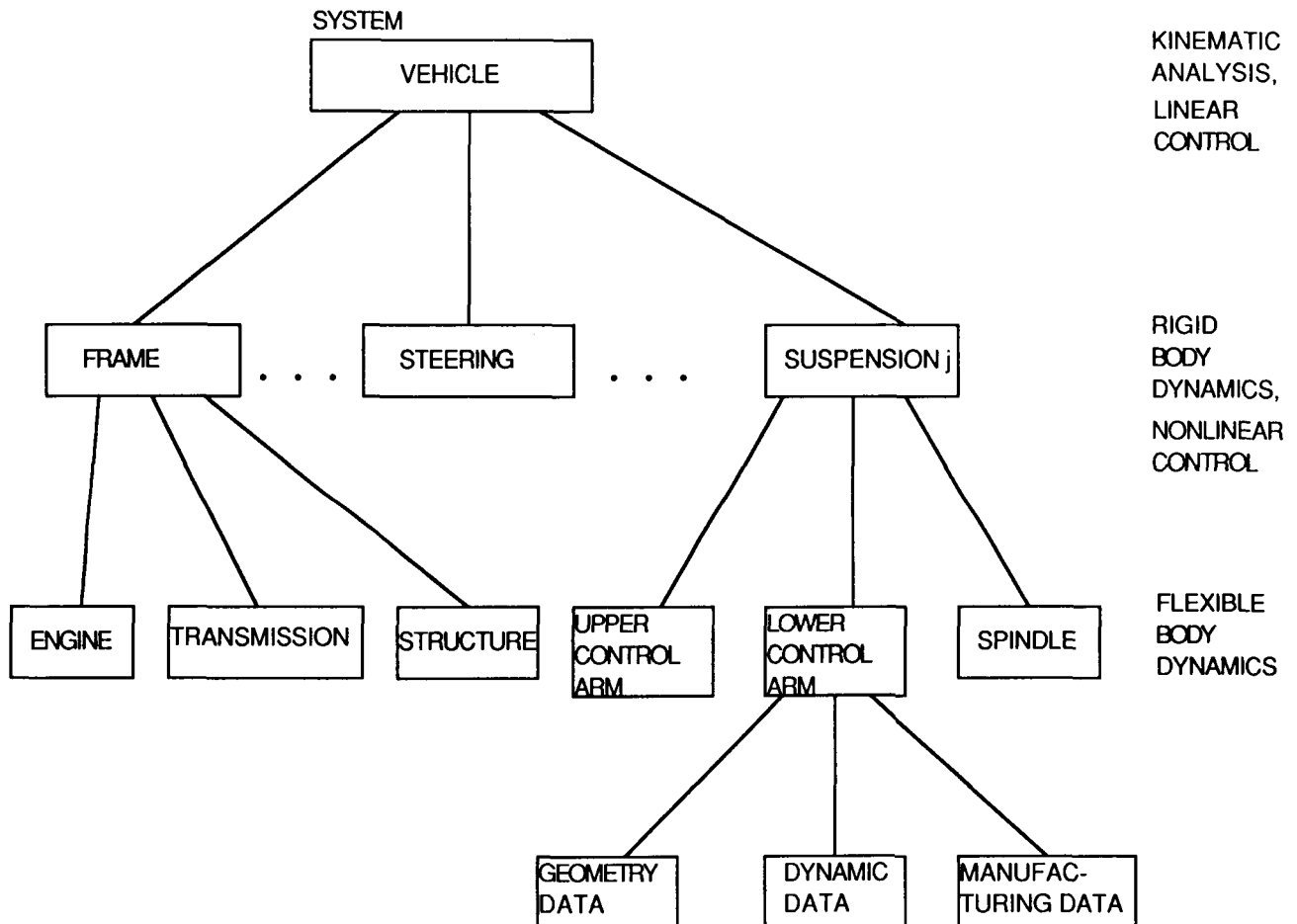


TOP DOWN DATA BASE DESIGN

The main criteria for developing the data base was to provide sufficient flexibility for extensions that may be required for later developments. To achieve this goal, an object oriented design methodology that is based on the object modeling technique was used to develop the data base (ref. 7). In an object oriented system, the external specifications are clearly separated from the internal implementation. A set of data and operations is associated with each object and related objects are grouped to facilitate reuse for similar code. A hierarchical organization of objects allows for top down design and utilization of the data base. That is, after the data associated with the highest level object is obtained, analysis codes that utilize these data may be executed, which in turn may create data associated with lower level objects.

As an example consider the figure below in which the data base for a vehicle is outlined. The highest object is the vehicle itself, which is an aggregation of the suspension subsystems, the frame, the steering mechanism, and many other objects. Even initial data associated with this object level hierarchy is sufficient for a crude system simulation, based on a rigid body dynamics model. If a more precise system simulation is required, the objects must be refined. For example, if the flexibility of a component must be considered, the object must again be refined. Now each object consists of subobjects for geometry data, dynamic data, and manufacturing data.

APPLICABILITY OF ANALYSIS TOOLS IN TOP DOWN DESIGN

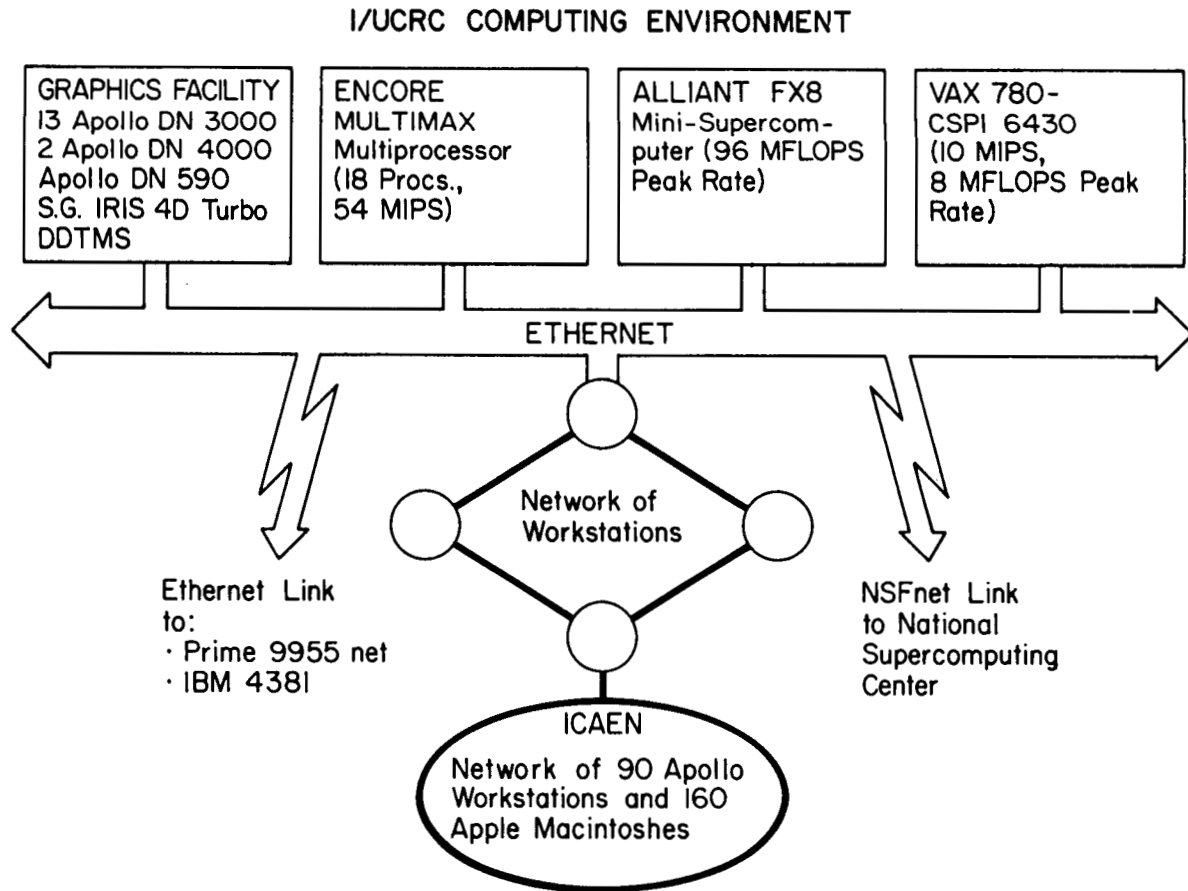


THE NETWORKING ENVIRONMENT

The analysis and design optimization system will be developed to handle local data bases that will assist designers in evaluating trial designs. Local area networks are being used to connect workstations, parallel computer(s), supercomputers, and other hardware, so computational tasks can be effectively distributed. Emerging tools such as user transparent distributed processing on workstation networks is being exploited.

The underlying software for such a network is (1) the Network File System (NFS) (ref. 2) that allows file access over a network of heterogeneous computers; (2) the Network Computing System (NCS) (ref. 3) that allows for distributed remote procedure calls; and (3) the X-Window system that allows creating windows and controls window management remotely (ref. 4).

A computer network is often composed of three components: a compute server, a data base server, and a user interface server. In the figure, the network environment at the Center for Simulation and Design Optimization at the University of Iowa is shown. As the computational server that will execute compute intensive tasks, such as finite element analysis or system dynamic simulation, the ALLIANT FX/8 and the ENCORE MULTIMAX are being used. As the data base server for long term data base storage, the VAX 11/780 is now being utilized. All user interface, modeling, and evaluation tools will be executed on a network of APOLLO workstations. For high performance graphics and real time animation, an IRIS 4D is being utilized.



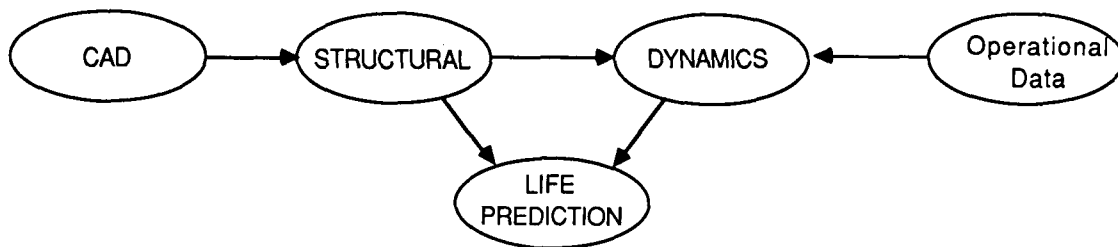
RUN STREAMS

The term "run streams" refers to a standard or user defined sequence of IAC provided operations. A run stream is often designed to handle a class of problems that consists of a number of selectable options and variations, rather than a rigidly predefined and automated process. An engineer-in-the-loop mode of operation is therefore possible and encouraged.

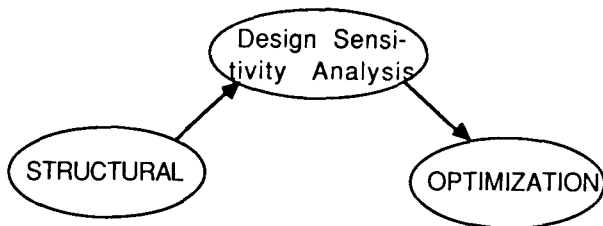
The figure illustrates several standard run streams that have been defined to provide structural/system dynamics/controls/optimization capabilities. Besides the standalone (uncoupled) operation of each technology or major technical module, only the stress history and fatigue analysis run stream has been implemented. The other implementations are under development.

The IAC facilitates the flow of data between different modules or between a module and the user, by providing a central data base storage area, standard data structures and formats such as relations and arrays, and data management tools. Multi-user concurrent access to the data base is supported. The IAC allows cataloging of structured and unstructured data base files and, with each data structured catalog entry, textual information such as keywords and data titles and pointers to text files can be defined.

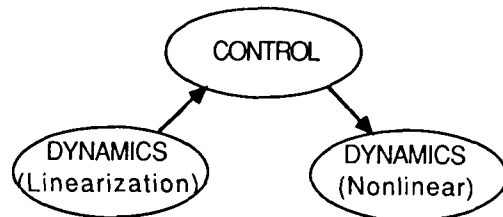
(I) LIFE PREDICTION RUN STREAM



(II) STRUCTURAL OPTIMIZATION RUN STREAM



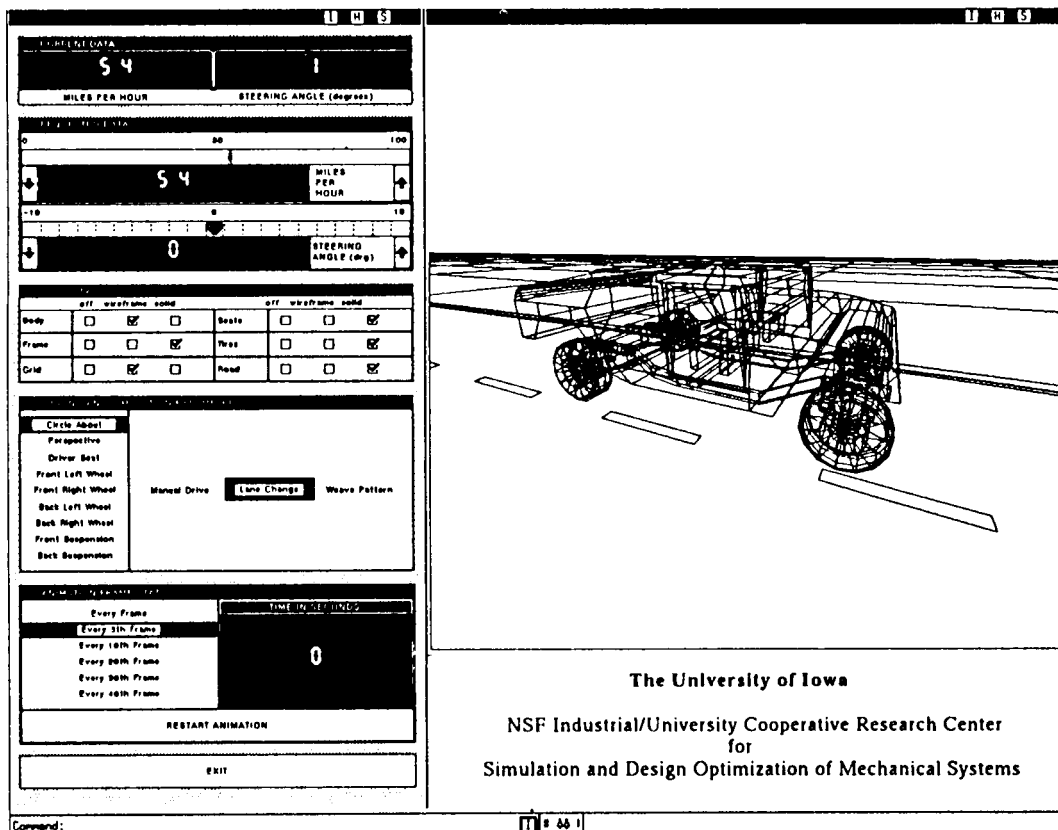
(III) CONTROL RUN STREAM



THE GRAPHICAL USER INTERFACE

A challenge for the analysis and design optimization system is to develop user interfaces that help the journeyman engineer take advantage of the network system (refs. 8 and 9). That is, the network system can be used by nonspecialist project engineers and will be flexible enough to fully exploit the network computing system (refs. 10 and 11). The overall integrated networked simulation and design optimization system will be set up in such a way that the experienced user can take full advantage of networks, whereas the inexperienced user need only know the concept and will employ a user friendly interface and support system to guide him through his applications.

The user interface is a multi-windowing system that executes on a workstation, where a variety of options will be laid out in a menu system and selections can be chosen by a mouse. The user interface menus are based on a user interface command language that (1) activates simulation, modeling, and evaluation tools and (2) provides numerical and graphical access to the data base.



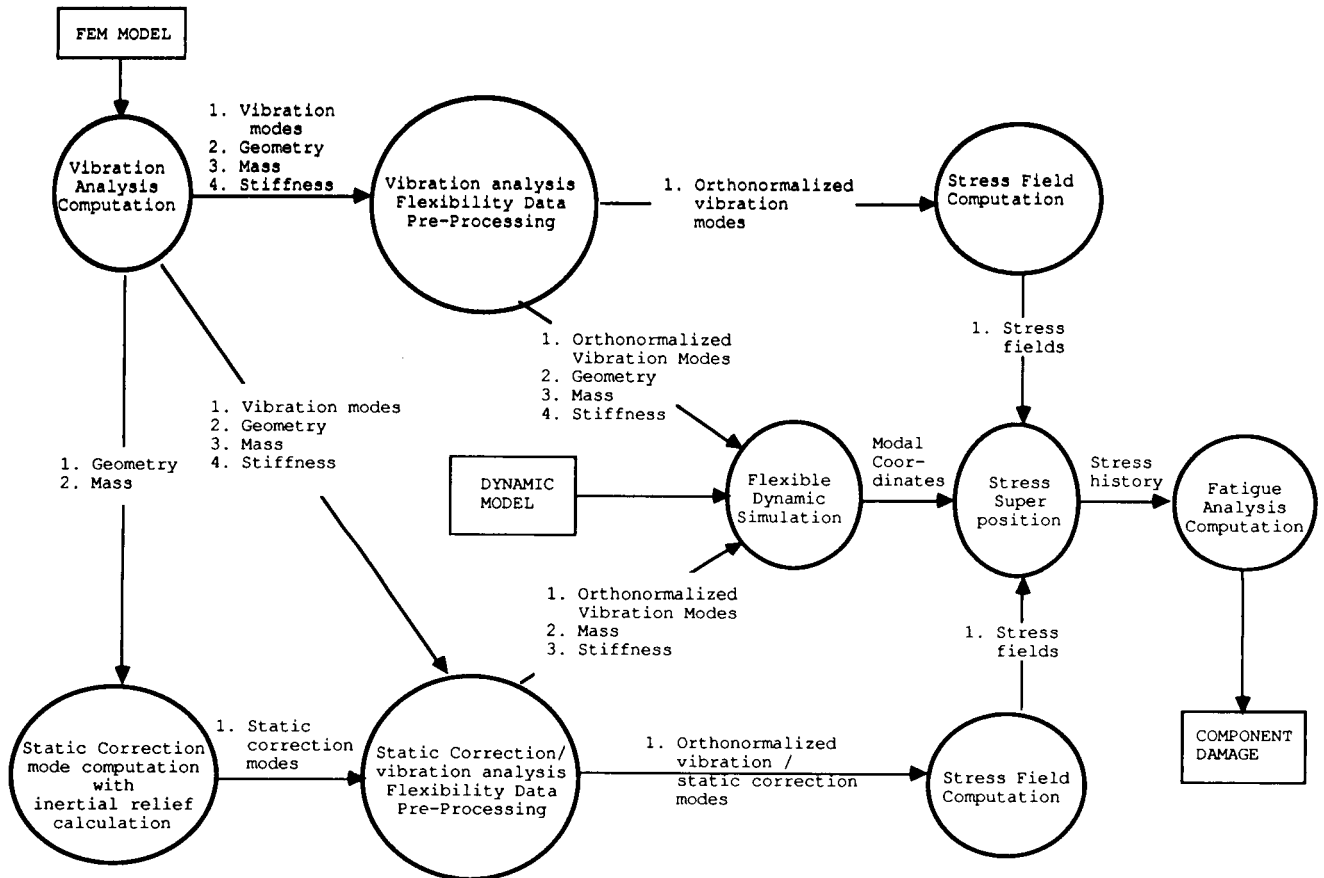
ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

THE FUNCTIONALITY OF THE LIFE PREDICTION RUN STREAM

The data flow for computation of mechanical system component loads, stress histories, and fatigue life prediction is shown in the figure. The method is based on a coupled gross motion-flexible body dynamic simulation model, as described in refs. 12, 13, and 14.

Each individual component that is represented as a flexible body in the mechanical system must be identified. Either vibration mode or vibration modes combined with static correction modes are computed, using the finite element method, to represent the flexibility in the individual system components. Flexibility data preprocessing prepares the output data to be used in the dynamic simulation. A combination of rigid and flexible components may be used in the dynamic simulation of the mechanical system. Large displacements occur between points on different components, but linear elastic theory is adequate to describe the deformation of individual components. The dynamic simulation computes the loads on the individual components and the contribution of each deformation field on the total deformation of the system at each time step. After the stress fields associated with the individual vibration and static correction modes has been computed, they can be superposed according the contribution each deformation field had in the flexible deformation of the bodies in the dynamic simulation. This will provide a stress history that then may be used to predict the fatigue life of individual components. The data input and output for each individual component is defined in the figure.

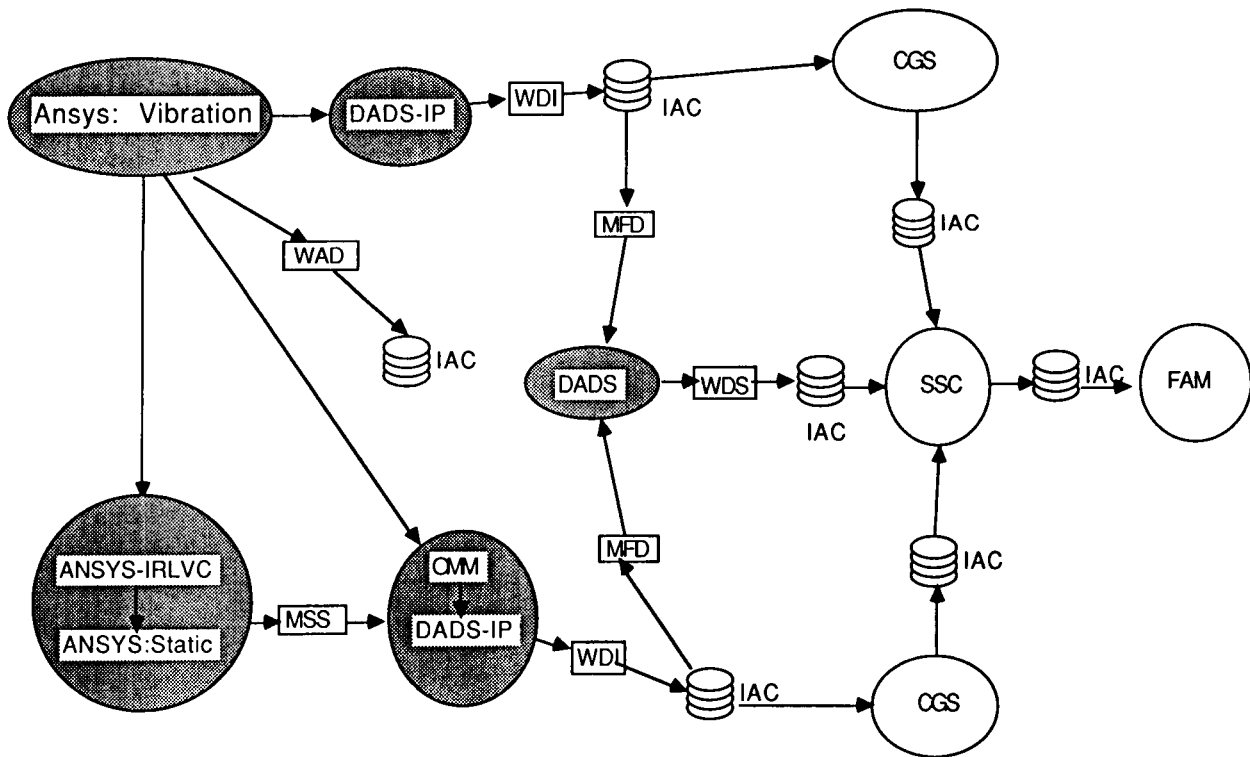
Dynamic Stress and Fatigue Analysis Computation



THE IMPLEMENTATION OF THE LIFE PREDICTION RUN STREAM

Based on the data flow described above, a run stream for the computation of stress history and fatigue life of mechanical systems is shown in the figure. For vibration and static correction mode computation, the ANSYS finite element code was used. Relevant data from the ANSYS output file are written to the IAC data base. Data needed from ANSYS for the intermediate processor are read directly from the ANSYS output files. The output from the intermediate processor is written to the IAC data base. For the dynamic simulation function, the DADS code was utilized. Because DADS is a code in which only the executable is available, all data to and from the data base to the code must go through interface codes. One separate code was written and implemented in the IAC for (1) computing stress fields for a given displacement field, (2) computing stress field superposition, and (3) fatigue analysis. Because these codes were written by the Center, communication between the codes and the IAC data base was direct and no interface codes were needed.

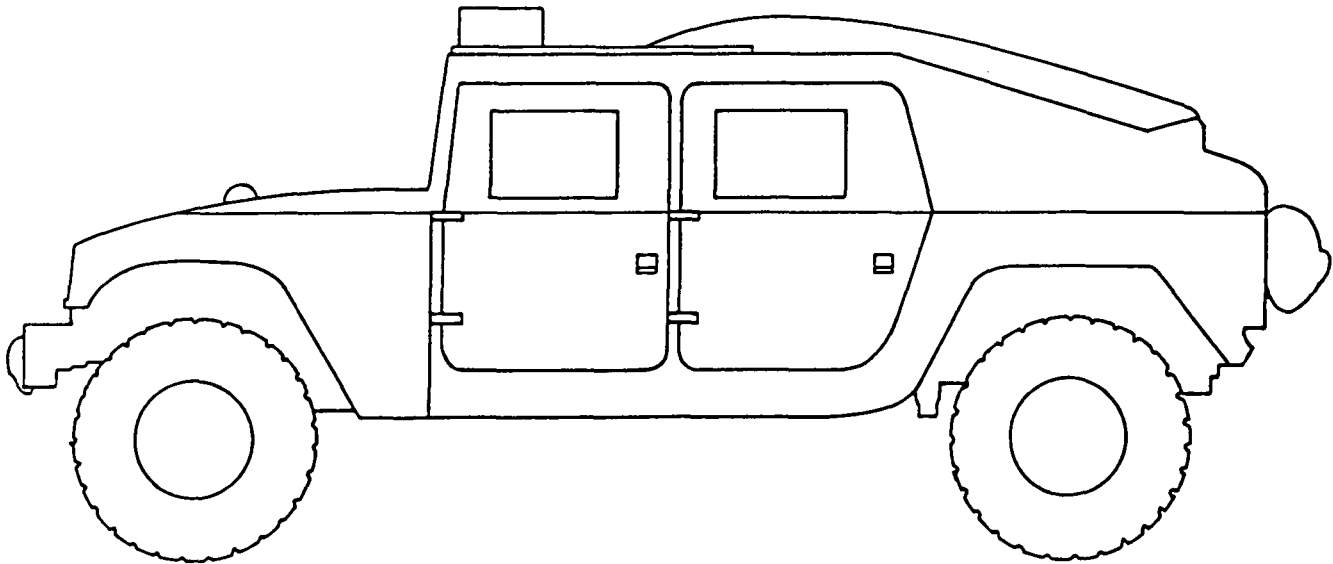
PROGRAM ORGANIZATION



THE HMMWV

The High Mobility Multipurpose Wheeled Vehicle (HMMWV) is used to demonstrate the applicability of the integrated, interdisciplinary system. It is 4.57 meters in length, 2.16 meters in width, and 1.76 meters in height and has a mass of 2340 kg. Bodies included in the dynamic model are the frame (including the nonstructural engine, transmission, and cab masses), control arms, and wheel assemblies. The HMMWV has four double-A-arm suspensions, each with two control arms and a wheel assembly. Hence, there are a total of thirteen bodies in the spatial model. Two spherical joints connect each control arm to beam elements that are attached to the chassis to represent bushing effects. Each control arm and the associated wheel assembly is connected by a spherical joint. In addition, there are distance constraints between each wheel assembly and the chassis, both in the front and in the rear, to represent steering tie rods.

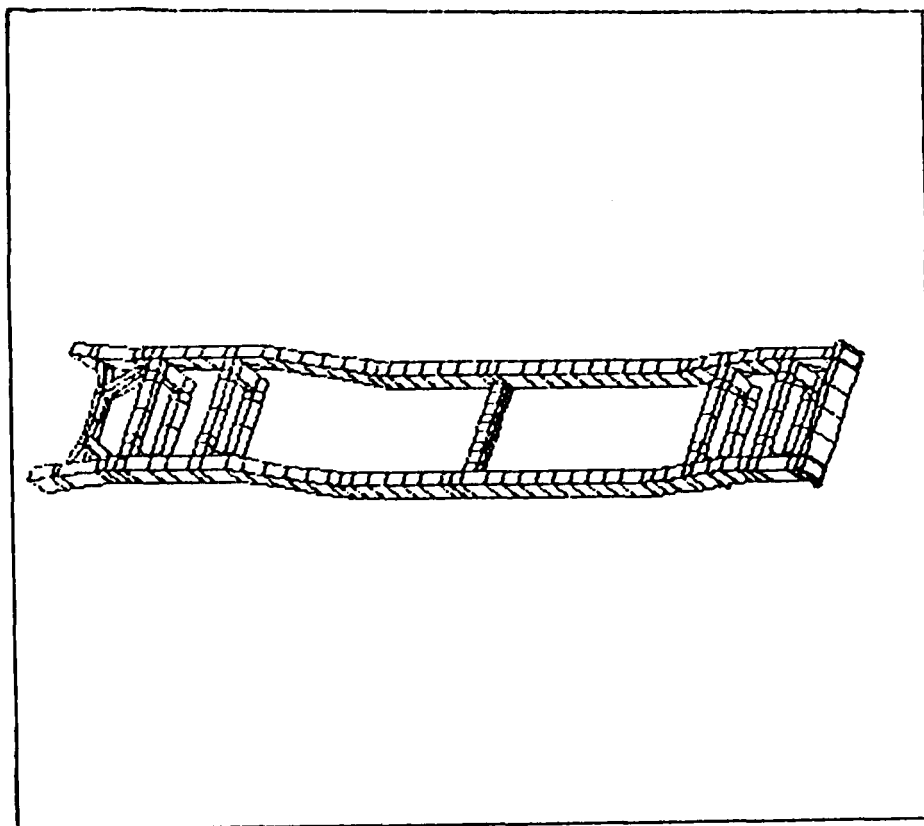
The flexibility of the chassis is represented using 10 vibration modes. Flexible body simulations are performed with the vehicle traversing obstacles and rough terrains to account for use of the vehicle. Loads induced by ground roughness are transmitted through the tires and suspension subsystem to the chassis. A spatial dynamic simulation of the vehicle over multiple ground profiles is carried out using the DADS code to determine the deformation of the frame at each time step.



MODE ANALYSIS

The vehicle model consists of a frame and nonstructural masses. As illustrated below, the frame of the vehicle is comprised of two side rails, four suspension crossmembers, one transmission crossmember, two bumpers, and two braces. Nonstructural masses are attached to the frame to account for the engine-transmission and the cab. The engine-transmission is supported by three beams, to account for inertia loads that act on the frame, due to acceleration of the engine-transmission. Similarly, the cab is attached to the frame by six body mounts that are represented as beams.

The ANSYS finite element analysis program is employed for stress analysis. The generalized mass element in ANSYS is used to represent masses and inertias of the engine-transmission and the cab. The generalized mass element has three translational and three rotational degrees of freedom. Masses and rotary inertias are thus specified at nodal points that correspond to centers of mass of nonstructural masses. Beam-truss elements, with six degrees of freedom per node (Element types 4 and 44 in ANSYS), are used for modeling the side rails, crossmembers, bumpers, and braces. Material property data for AISI C1020 steel is input. A finite model of the frame with 56 elements, 49 nodes, and 294 degrees of freedom is adequate for this elementary structural model. Both the theoretical foundation and the implementation do not place any limitation on the model dimension. A refined FEM model of the frame is shown below.

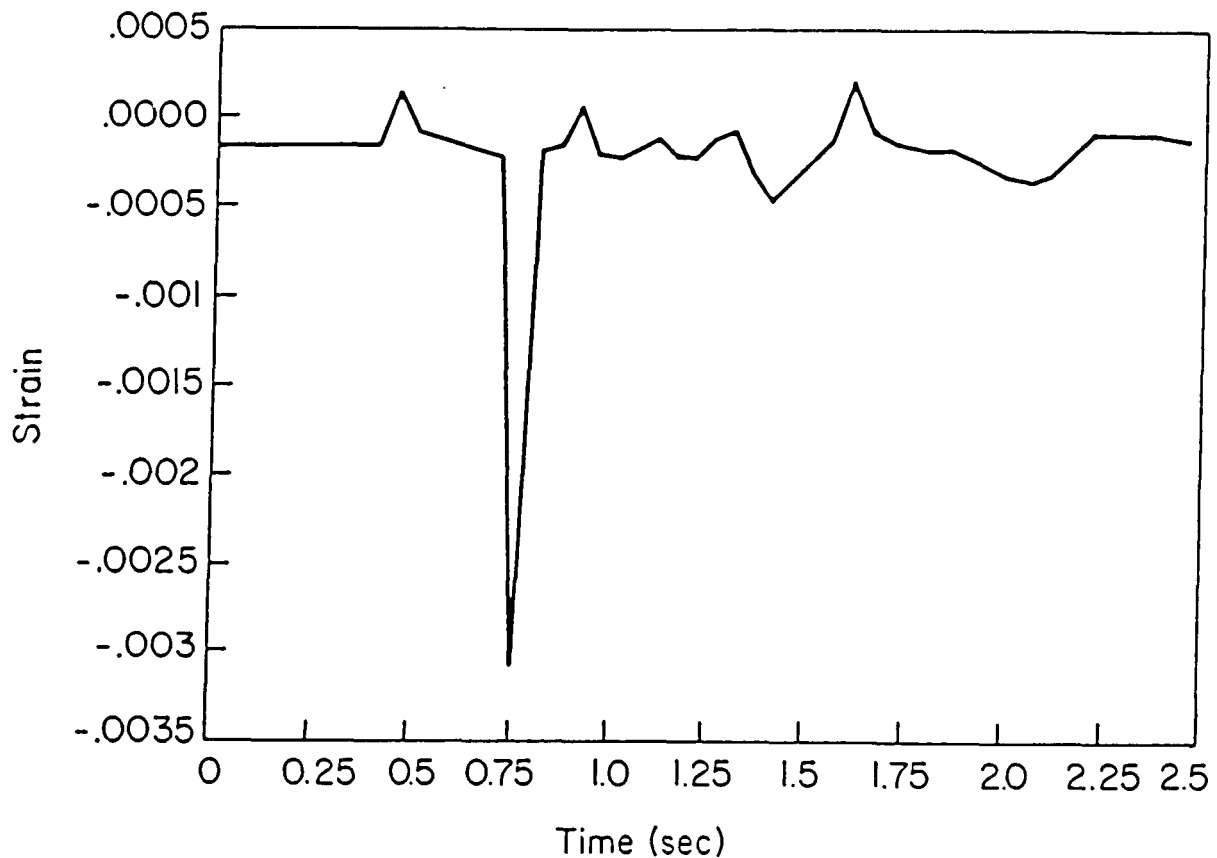


DYNAMIC STRESS HISTORY

As a typical dynamic simulation environment, the vehicle traverses a 4 in. high one sided bump with a speed of 22.5 mph. Because of this unsymmetrical terrain, the dynamic response and stress histories in the vehicle are unsymmetric. The simulation is carried out for 2.5 sec on a reporting time grid of 0.05 sec. Stress data are calculated for each node in each element.

As an example, the nominal strain history in the rearmost suspension crossmember, close to the lower corner bracket connection, is given. In the physical component, there is a hyperbolic notch. Large internal forces that vary with time occur here, due to large suspension and tire forces that are transmitted through the crossmember as the vehicle traverses the bump.

The insights and estimates of stress histories obtained in this analysis are of significant design value because they give the designer a clear idea of the effects of the coupled gross motion and flexible body dynamics on the system. This allows for improvement in design before building a prototype and testing, which significantly speeds up the design cycle.



CONCLUSIONS

The concept of an advanced integrated, networked simulation and design system has been outlined. Such an advanced system can be developed utilizing existing codes without compromising the integrity and functionality of the system. An example has been used to demonstrate the applicability of the concept of the integrated system outlined here.

The development of an integrated system can be done incrementally. Initial capabilities can be developed and implemented without having a detailed design of the global system. Only a conceptual global system must exist. For a fully integrated, user friendly design system, further research is needed in the areas of engineering data bases, distributed data bases, and advanced user interface design.

The integrated system must be based on

- An integrated command processor and data base management system
- A network environment
- A graphical window based user interface

We concluded that

- Existing codes can be effectively utilized
- An interdisciplinary analysis system can be developed incrementally

REFERENCES

1. Dopker, B.: Developments in Interdisciplinary Software for Mechanical System Simulation and Design. *Engineering with Computers*, vol. 4, 1988, pp. 229-238.
2. Network File System (NFS), Sun Microsystems Inc., (Mountain View, California), 1987.
3. Network Computing System (NCS), Apollo Computer Inc., (Chelmsford, Massachusetts), 1987.
4. Scheifler, R.W.; and Gettys, J.: The X Window System. *ACM Transactions on Graphics*, vol. 5, 1986, pp. 79-109.
5. Vos, R.G.; Walker, W.J.; Beste, D.L.; Price, G.A.; Young, J.P.; and Frisch, H.P.: Development and Use of an Integrated Analysis Capability. *AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference* (Lake Tahoe, Nevada), May 1983.
6. Vos, R.G.; Beste, D.L.; Gregg, J.; and Frisch, H.P.; and Sanborn, J.A.: Integrated Analysis Capability (IAC), Level 1.5. COSMIC Library, University of Georgia, (Athens, Georgia), 1985.
7. Blaha, M.R.; Premerlani, W.J.; and Rumbaugh, J.E.: Relational Database Design Using an Object-Oriented Methodology. *Communications of the ACM*, vol. 31, no. 4, 1988, pp. 414-427.
8. Woodward, W.S.; and Morris, J.W.: Improving Productivity in Finite Element Analysis Through Interactive Processing. *Some Perspectives on CAD/CAM in Mechanical Engineering*, (Ed.: Perrone, N.; and Magrab, E.B.), ASME PVP-vol. 87, 1984.
9. Wu, J.K.: Graphics-Based Mechanical System Simulation and Design Preprocessing. *Proceedings of the 6th National Conference on University Programs in Computer Aided Engineering Design and Manufacturing*, (Atlanta, Georgia), June 1988.
10. Dubetz, M.W.; Kuhl, J.G.; and Haug, E.J.: A Network Implementation of Real-Time Dynamic Simulation with Interactive Animated Graphics. *Design Automation Conference*, (Orlando, Florida), September 1988.
11. Dubetz, M.W.; and Haug, E.J.: Real-Time Dynamics Simulation - A Design Optimization Tool. *Design Automation Conference*, (Orlando, Florida), September 1988.
12. Liu, T.S.; Haug, E.J.; and Dopker, B.: A Dynamic Simulation Approach to Machine Component Life Prediction. *Design Automation Conference*, (Orlando, Florida), September 1988.
13. Stephens, R.I.; Dopker, B.; Haug, E.J.; Back, W.K.; Johnson, L.P.; and Liu, T.S.: Computational Fatigue Life Prediction of Welded and Non-Welded Ground Vehicle Components. *Proceedings of the SAE Passenger Car Meeting*, (Dearborn, Michigan), October 1987.
14. Liu, T.S.: Computational Methods for Life Prediction of Mechanical Components of Dynamic Systems. Ph.D. Thesis, The University of Iowa, (Iowa City, Iowa), December 1986.